OUR REF AN-C24/3

Greenbank Electronics

DATE 14-11-80 Revised 21-2-81, 6-6-81, 27-1-89 (address changed)

CONVERSION OF KEMITRON PP-2 BOARD TO PROGRAM 2508/2516/2532

Introduction

The well known 2708 EPROM is gradually being rendered obsolete by the newer types which are now becoming cheap enough to consider for new designs..

The newer EPROMs to which we refer are given various numbers by various manufacturers but we are going to list them as follows:-

1K	:	2508
2K	:	2516
4K	:	2532
8K	:	2564

These all share the following benefits:-

① Only a single 50 ms TTL level pulse is required to program them (after a fixed 25V programming voltage has first been applied to the Vpp pin).

② They only need a single +5V supply for reading.

③ The EPROM can be programmed in part or in whole, entirely at the user's option.

The corresponding drawbacks to the 2708 and TMS 2716 EPROMs are as follows:-

① The programming pulse has to swing from 0 to +25V; the whole set of locations need to be pulsed in turn and the procedure repeated 100 or more times for the data to 'stick'.

② Three supplies are needed (+12V, +5V, -5V) for reading.

③ It is not permitted to program only a part of 2708 or TMS 2716.

460, NEW CHESTER ROAD, ROCK FERRY, BIRKENHEAD, MERSEYSIDE. L42 2AE
TELEPHONE: 051-645 3391.

Design of a programmer:

There are many differing views on how an EPROM programmer should be designed. The existing Kemitron board for 2708s (the 'PP-2' board) has proved very acceptable, and we think that the 'memory-mapped' technique has a lot to recommend it:

Software is fairly easy to write – as the EPROM is memory-mapped it is simply a matter of writing data to it just like any other memory device, (having first switched on the programming voltage), applying the programming pulse, then moving on to the next location.

As the EPROM is memory-mapped, it is very easy to verify that a location is empty before it is programmed, and that data is correctly stored after programming. (These two tests are optional of course, depending on the user's needs).

It is very handy to have an empty socket in the design, to enable straight copies to be made for 'back up' or other purposes.

There is a 'control byte' which controls the 25V supply and the TTL pulses, and ideally we think this should be in the I/O space, which would be the case in a new design of our own. However, when the time came for us to build a new 2516 EPROM programmer, we ourselves found it much easier to adapt the existing PP-2 board than to build one of our own. It turns out to be easier, due to the design of the PP-2 board, to locate the control byte in the memory space, so we have temporarily had to forgo our preference for locating it in the I/O space.

This note describes some of the modifications made, and is presented to aid other users who want to follow the same path as ourselves.

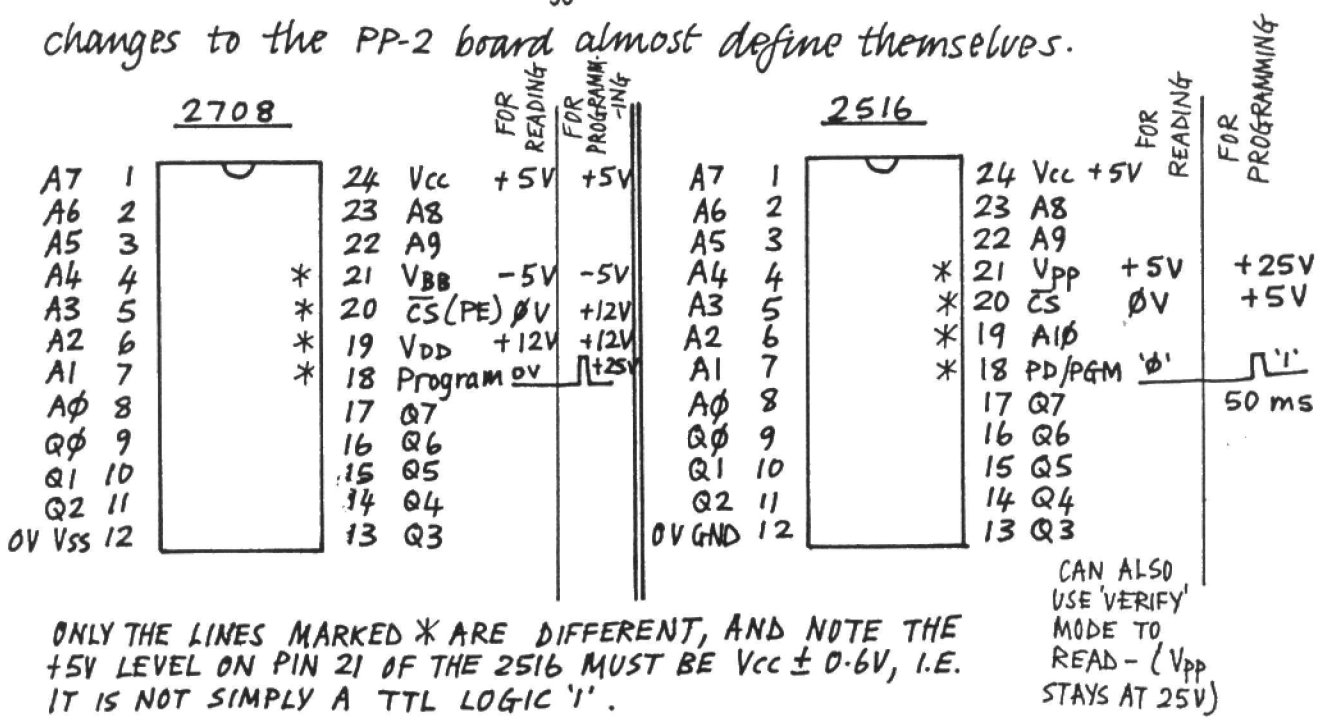
What about 2508, 2532, 2564?

In what follows, only the 2516 programmer will be described. The 2508 and 2532 are very similar and need only a few minor changes from the 2516 – it is left for the user to consult the appropriate data sheets to see what is involved. (The 2564 is more difficult as it has 4 extra pins and so won't plug in a 24-pin socket, and it needs an extra latched address line, which is not directly available on the PP-2 board as it stands).

In practise the programmer need only cater for the 2516 and 2532 at the moment - the 1K 2508 ironically usually costs more than the 2K 2516 due to the reduced price competition for the smaller device, and the 8K 2564 is very scarce and expensive at present.

Comparison of 2708's and 2516's

The following diagram shows the main differences between the 2708 and the 2516, and once the differences have been understood, the changes to the PP-2 board almost define themselves.



ONLY THE LINES MARKED * ARE DIFFERENT, AND NOTE THE +5V LEVEL ON PIN 21 OF THE 2516 MUST BE Vcc ± 0·6V, I.E. IT IS NOT SIMPLY A TTL LOGIC 'I'.

Constructional Points

The following diagrams show the circuit and layout of the board before and after modification. The precise details of the modifications are not included, since they become very repetitive to list, but the main points to bear in mind are that the existing connections should first be removed by cutting tracks or removing through hole links before the new connections are made, and also that it is usually possible to leave handy termination points for the new wires, by making any track cuts intelligently.

Address Decoding

The addresses below have been chosen. (The decisions involved have not been taken lightly and the reader is directed to our note, Reference No. AN-C23 for the details of the whole memory map and some further comments).

EMPTY EPROM : Begins AØØØ up to 4K

'MASTER' OR SOURCE : Begins BØØØ up to 4K. (Except of course when a 2564 has been located at AØØØ - this will extend over the whole of pages A and B, and so the 'master' will have to be somewhere else.)

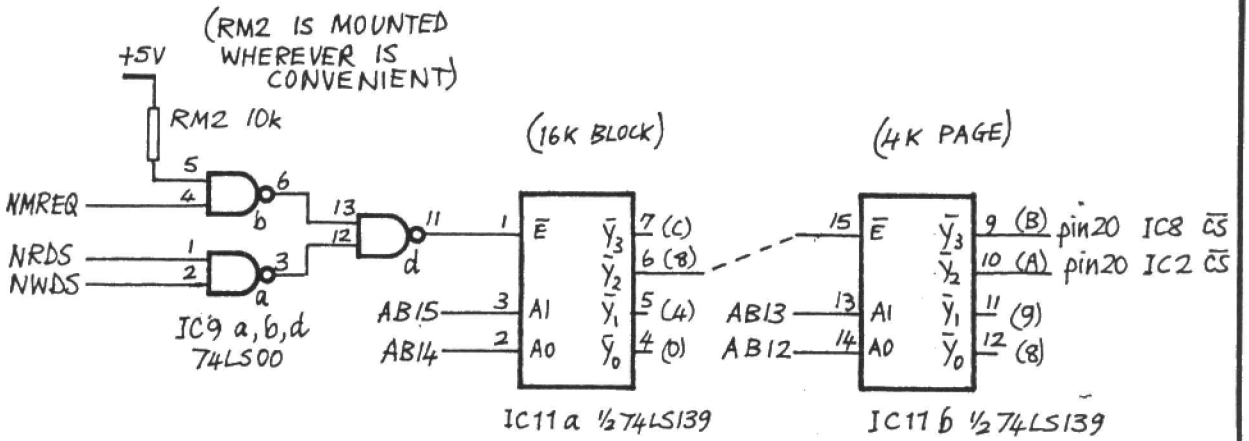
CONTROL PORT : Port FØ

One implementation to achieve the above, using the decoding chips already available on the board, is shown in the diagram on page 5.

An extra 'OR' gate pack type 74LS32 has to be added and a pull-up resistor.

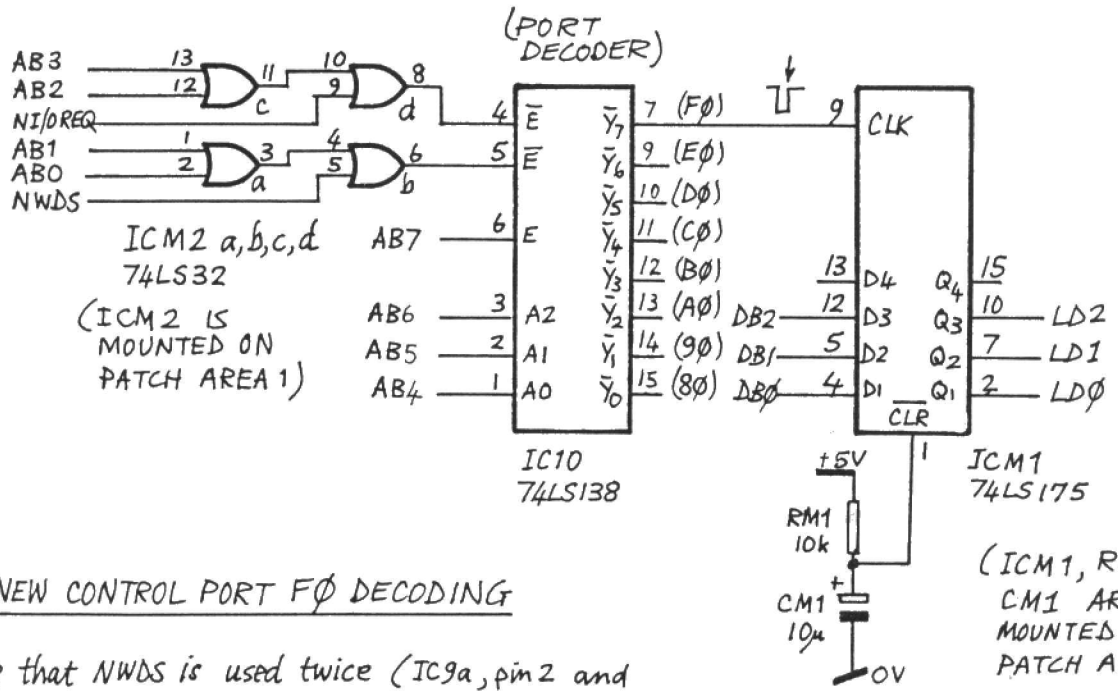
However while the circuit is being modified an extra change that can be carried out is to add a 74LS175 instead of the on-board 74LS75.

The reasons for this latter change are given on the next page of description, page 10.



MODIFIED ADDRESS DECODING

Note that $\overline{CS}$ IC8 and $\overline{CS}$ IC2, (ie. pins 20), are each enabled for the whole of the 4K page – this is to permit a future upgrade to the 2532 to be made, without the need to modify the address decoding again.



NEW CONTROL PORT FØ DECODING

Note that NWDS is used twice (IC9a, pin 2 and ICM2b pin 5) and so this card puts two loads on this bus line not one.

Also (NRDS is not brought into the above decoding; this is to prevent ICM1 latching erroneous (and destructive!) data in the event of a read Port FØ being inadvertently executed).

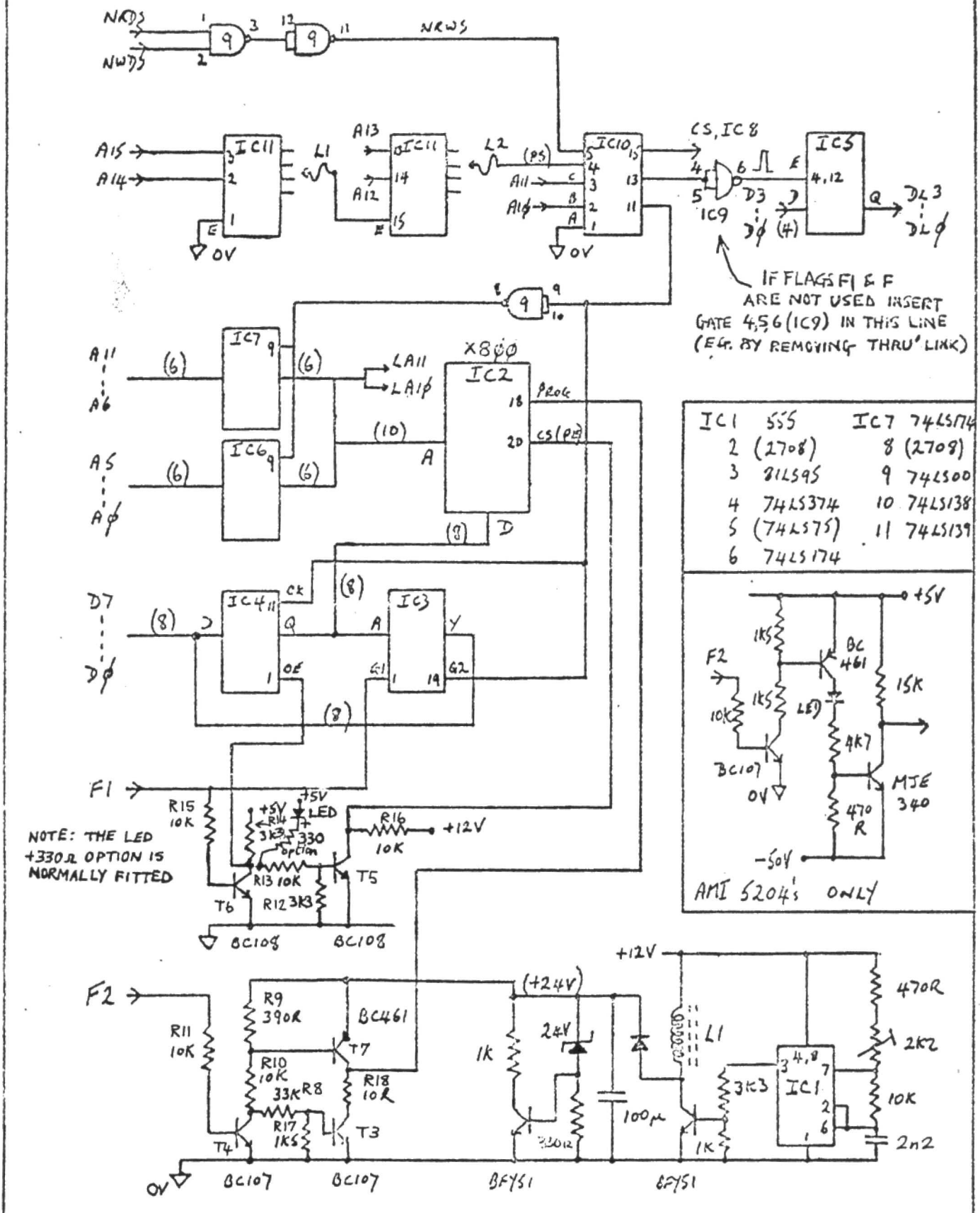
$\overline{Y}0$-$\overline{Y}7$ ON IC10 RENUMBERED 6-6-81

Drwn	D.M.P.
Date	21-2-81
Scale	–
	corrected 6-6-81

Greenbank Electronics

MODIFIED ADDRESS AND CONTROL PORT DECODING.

AN-C24/2 SHEET 5

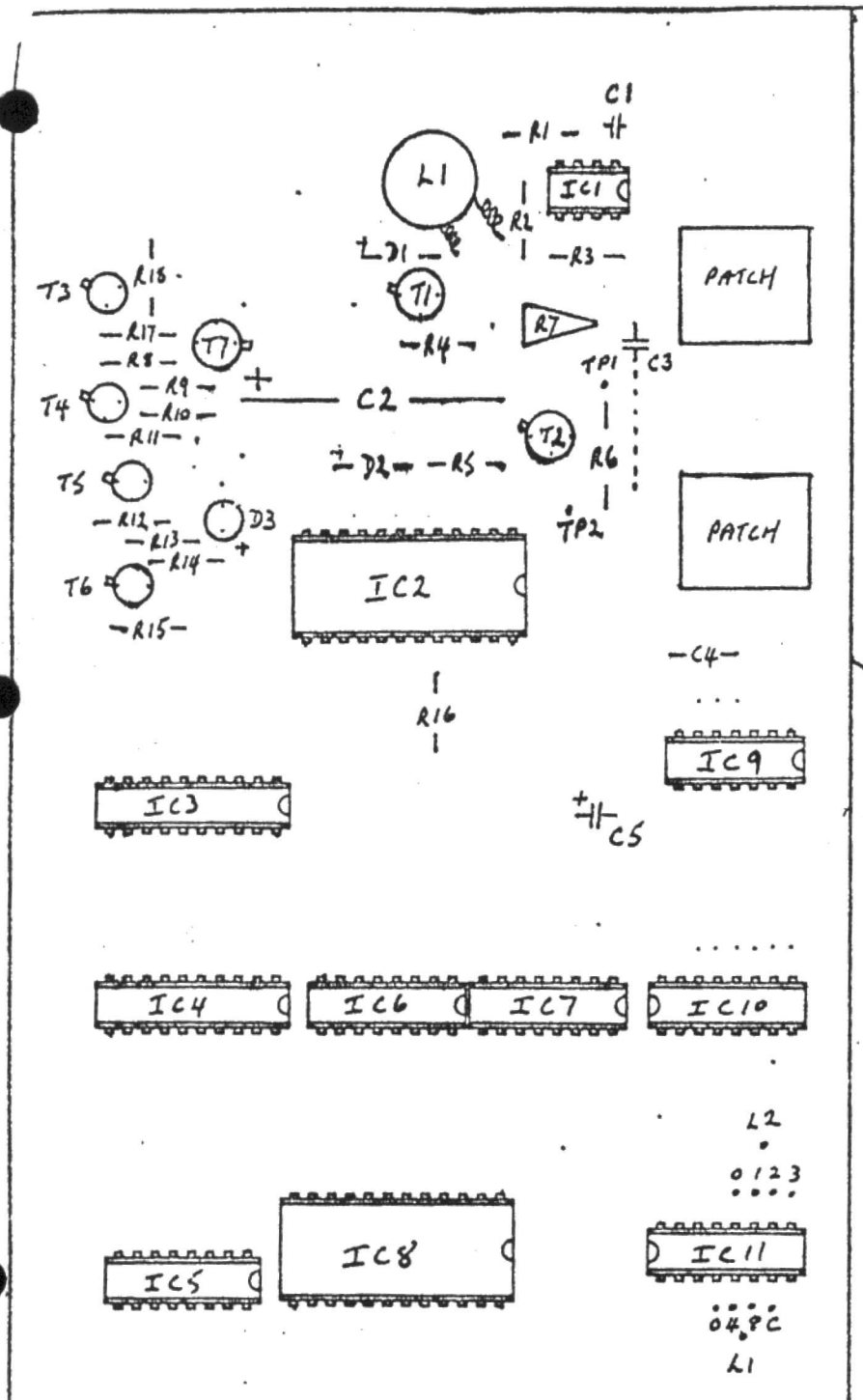


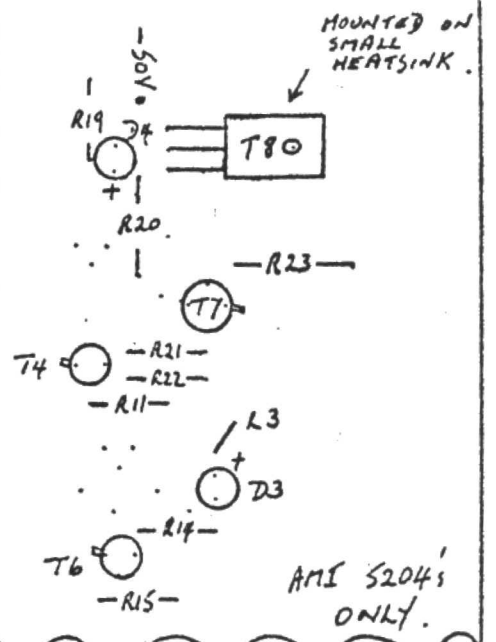
Kemitron Electronics

Drn. BSJ	Date 29/12/78

PP.2

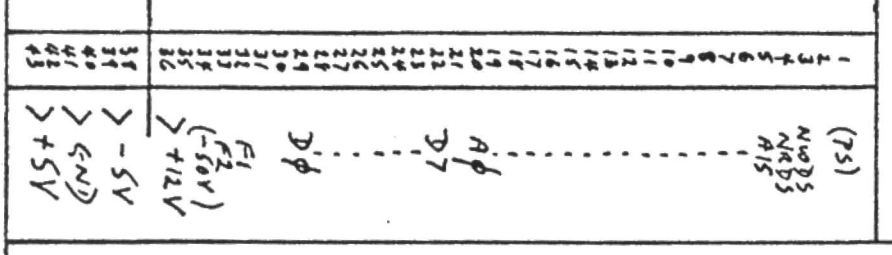
CIRCUIT FOR 2708's - THIS IS THE CIRCUIT DIAGRAM SUPPLIED WITH THE BOARD





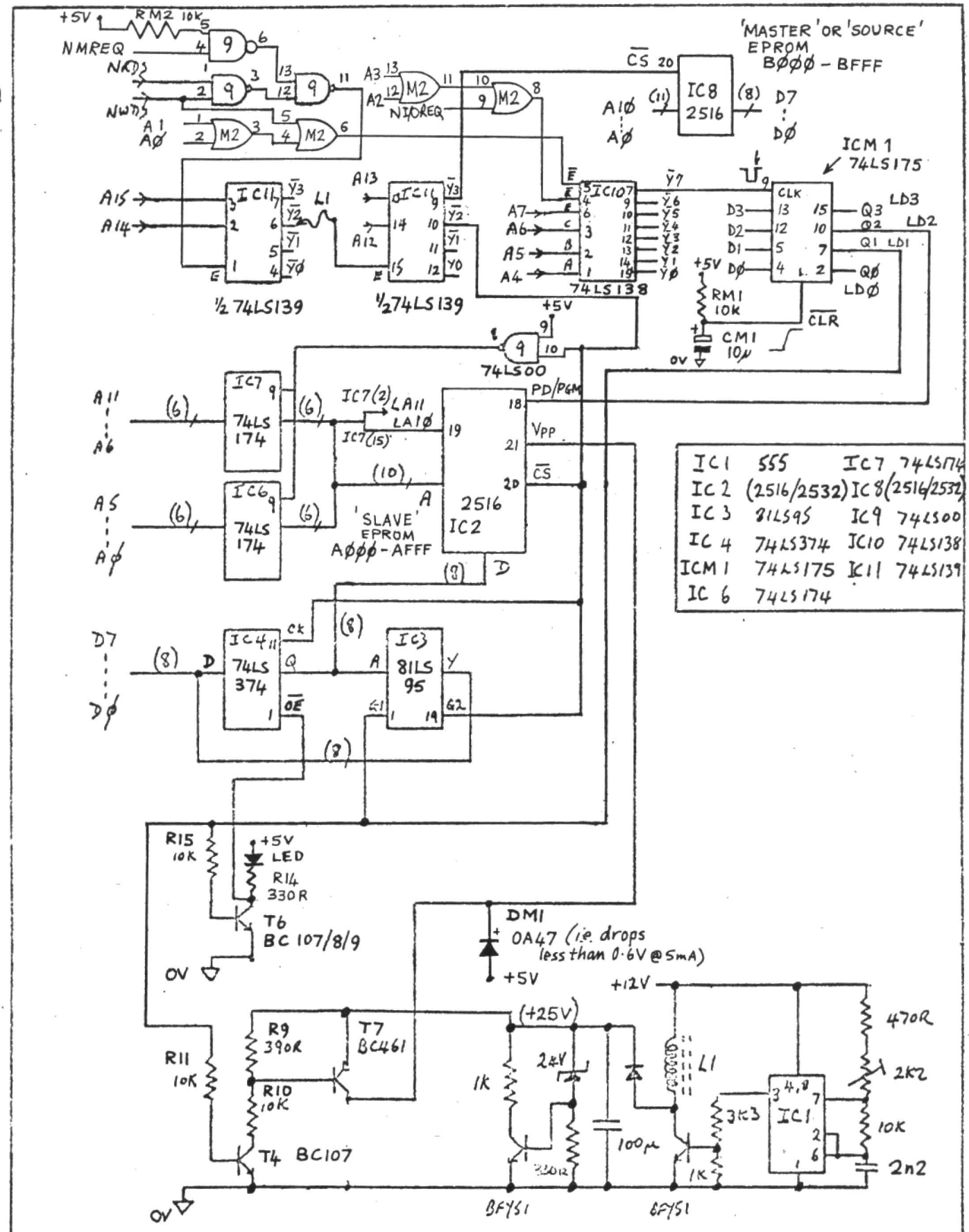
COMPONENTS

R1	3K3		R13	10K	
2	470R		14	330R (3K3)	
3	10K		15	10K	
4	1K		16	10K	
5	330R		17	1K5	
6	1K, ½W		18	10R	
7	2K2	PRESET	19	470R	
8	33K		20	4K7, ½W	
9	390R		21	1K5	
10	10K		22	1K5	
11	10K		23	15K	
12	3K3				
C1	2n2		C4	0·1μ DISC	
2	100μ, 40V		5	100μ 10V	
3	10μ, 16V		6	0·1μ DISC	
D1	IN4001		D3	LED	
2	BZY88, 24V		4	LED	
T1	BFY51		T5	BC107/8	
2	BFY51		6	BC107/8	
3	BC107		7	BC461	
4	BC107		8	MJE340	
IC1	555		IC7	74LS174	
2	(2708)		8	(2708)	
3	81LS95		9	74LS00	
4	74LS374		10	74LS138	
5	(74LS75)		11	74LS139	
6	74LS174				



Kemitron Electronics

Drn. PSD	Date 28/11/78	PP.2 COMPONENT LAYOUT FOR 2708's (AS SUPPLIED WITH CIRCUIT DIAGRAM - SHEET 5)



IC1	555	IC7	74LS174
IC2	(2516/2532)	IC8	(2516/2532)
IC3	81LS95	IC9	74LS00
IC4	74LS374	IC10	74LS138
ICM1	74LS175	IC11	74LS139
IC6	74LS174		

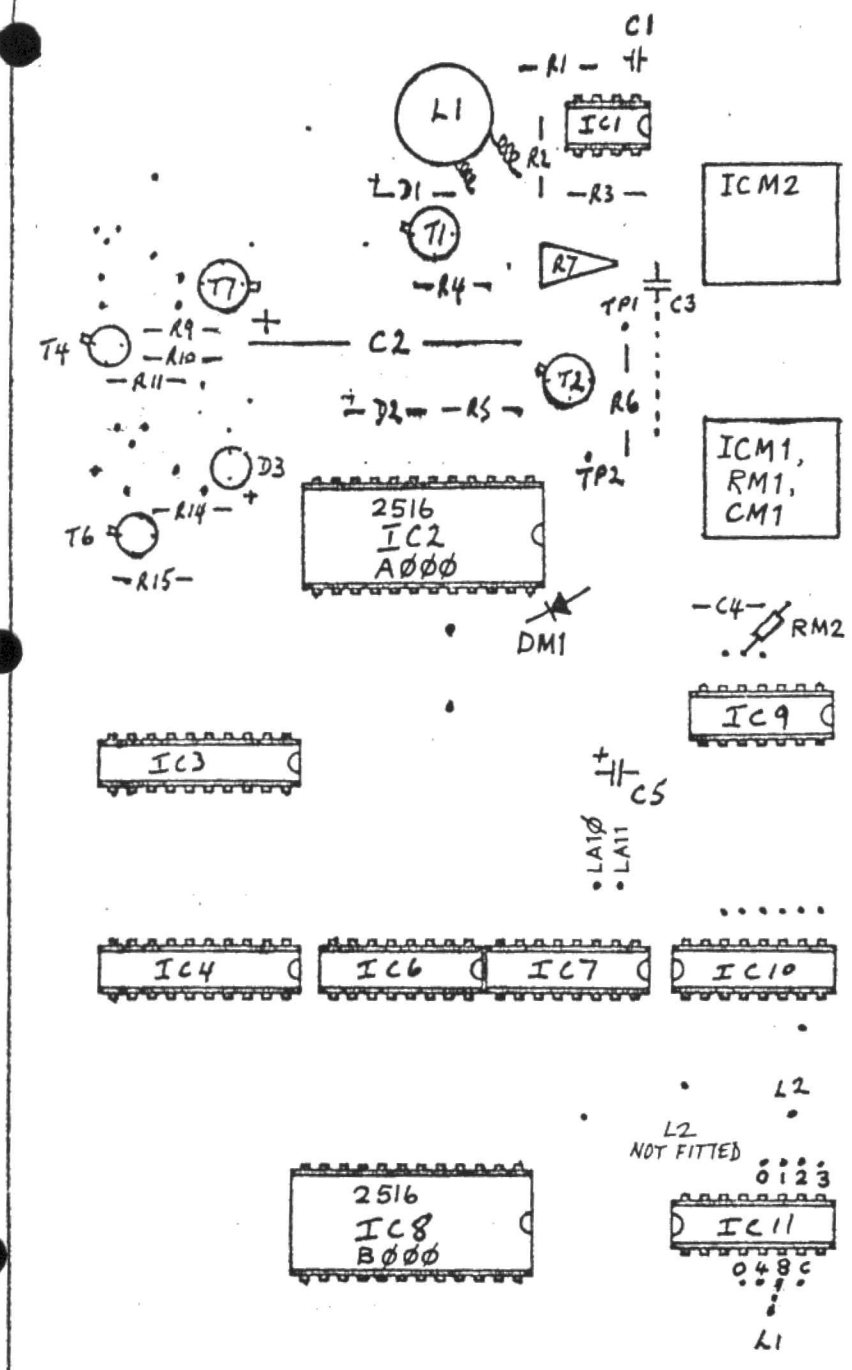
Kemitron Electronics

Drn. BSJ	Date 29/12/78
Mods. DMP	11/11/80, 21/2/81

PP.2

CIRCUIT DIAGRAM AFTER MODIFICATION FOR 2516's (OR 2532's WITH MINOR CHANGES).

MODED. 6/10/83.



COMPONENTS

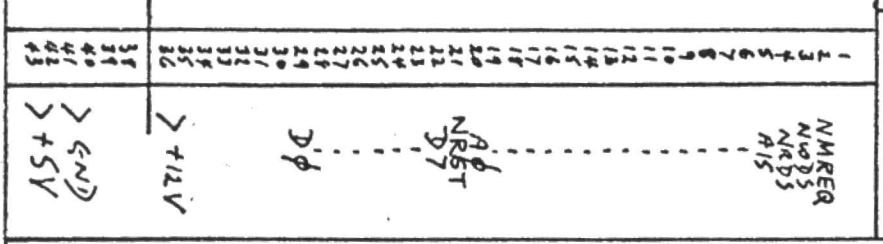
R1	3K3		
2	470R	14	330R (3K3)
3	10K	15	10K
4	1K		
5	330R		
6	1K, ½W		
7	2K2 PRESET		
9	390R		
10	10K	RM2	10k
11	10K	RM1	10k

CM1	10µ 16V		
C1	2n2	C4	0·1µ DISC
2	100µ, 40V	5	100µ 10V
3	10µ, 16V	6	0·1µ DISC

D1	IN4001	D3	LED
2	BZY88, 24V	DM1	DA47

T1	BFY51		
2	BFY51	6	BC107/8
		7	BC461
4	BC107		

IC1	555	IC7	74LS174
2	(2516/2532)	8	(2516/2532)
3	81LS95	9	74LS00
4	74LS374	10	74LS138
ICM1	74LS175	11	74LS139
6	74LS174	ICM2	74LS32



Kemitron Electronics

Drn. [illegible]	Date 28/11/78	PP.2 COMPONENT LAYOUT AFTER MODIFICATION (SEE SHEET 7 FOR CIRCUIT DIAGRAM)
Mods. D.M.P.	11/11/80, 21/2/81	'L2 NOT FITTED' 25-3-81

Control Word

Shown on the diagram at the foot of sheet 4 is a 4-bit flip-flop ICM1 (74LS175) which is extra to the original PP-2 circuit, and mounted on patch area 2. The 74LS75, (IC5 on the PP-2 circuit diagram), is not used by us as it does not have a 'clear' pin and so could switch on with random data, potentially causing damage to the EPROM it controls. At 'switch on' the resistor and capacitor network (RM1 + CM1) connected to pin 1 ('$\overline{CLR}$') of the 74LS175 holds it low long enough to cause all outputs to be 'ϕ', the safe condition.

If an ISBUS system is in use, there is already a suitable reset signal NRST on pin A21 of the edge connector, but it is worthwhile retaining the resistor and capacitor on the board in case the board is plugged in with the CPU card absent. (A 2-input 'AND' gate should be used to combine both signals.)

The data bus connections can conveniently be taken from the vacant IC5 position to ICM1.

The significance of the three outputs from ICM1 is as follows:-

LDϕ (pin 2) : IC2 +5V etc power ON/OFF (see note)

LD1 (pin 7) : IC2 +25V to pin 21 ON/OFF

LD2 (pin 10) : IC2 programming pulse.

Note: LDϕ is unused in this design but is reserved so that a future board could use this signal could make the IC2, IC8 and EPROM sockets 'dead' without switching the whole system off and possibly losing valuable data in RAM.

The use of the control word data is demonstrated in the example programs given at the end of this note.

(Users of INS 8060 (SC/MP II), of course can use their own flag lines instead of ICM1 but they are not considered here since they are not 'ISBUS' lines.)

+25V Supply

This is taken from the same ringing choke converter running from the +12V rail as is used in the standard PP-2 board. The use of +12V is legitimate as it is an ISBUS voltage, but ideally the circuit should be changed to run off +5V only, to suit a wider audience, but we have no details of the necessary changes.

The diagrams on pages 8 and 9 are based as closely as possible on Komitron's originals for the 2708 programmer so that you can see the extent of the changes simply by comparing both sets.

The diode DM1 is used to provide +5V (i.e. Vcc ± 0·6V) to pin 21 of IC2 when it is not being programmed, and therefore needs to drop less than 0·6V when it is carrying 5mA, and to withstand 20V reverse voltage. The only diode we have found which we can guarantee to meet this specification is a germanium type OA47, but in an emergency a IN4002 (e.g.) can be tried, as it comes very close to meeting the specification.

Other Details

Remember also to remove the +12V and -5V supplies from IC2 and IC8, and connect address line AB1Ø to pin 19 of IC8 (if it is a 2516 type). AB11 will have to be added as well for 2532s.

Similar connections, but this time the latched addresses (LA10, LA11) apply to IC2. These signals are already available on the PP-2 board:

LA1Ø : IC7 pin 15
LA11 : IC7 pin 2

Front Panel:

Our own suggestion is that this be a standard 2" wide type, with space for two zero insertion force sockets (one 'destination' one 'source').

Ribbon cable can be used to connect IC2 and IC8 positions to the front panel sockets after the board has been tested (e.g with 'solderon' pins temporarily on the board).

Except for the front panel the assembled board can be kept below 1" and so there is no need to waste one of the ISBUS socket positions.

The diagrams on the next two pages show the cutting dimensions we used, and the method for fixing the zero insertion force sockets.

Testing.

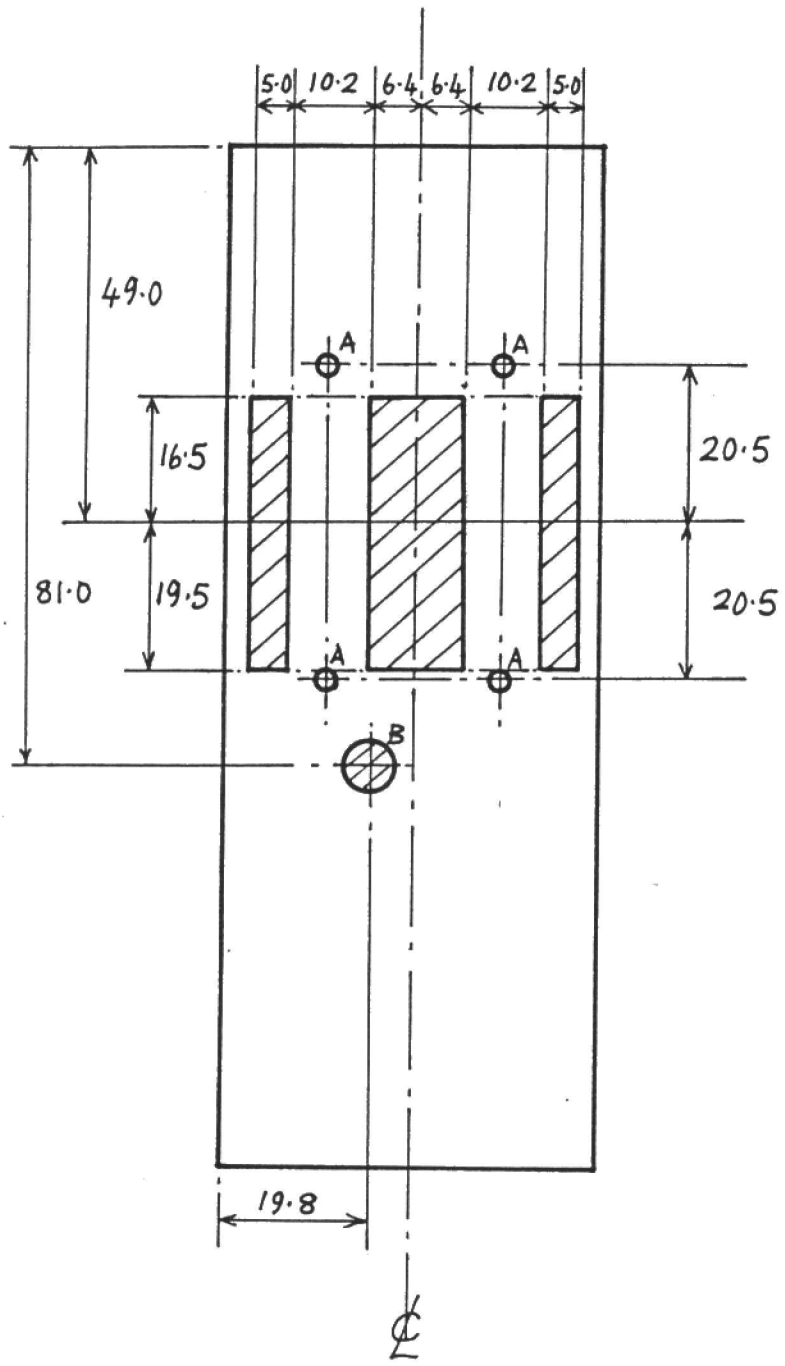
It goes without saying that an EPROM should not be plugged into its socket for the first time until an oscilloscope has been used to probe the various pins, especially the +25V and the programming pulses.

When the time comes to insert a 2516, it is suggested that only a few bytes be programmed since it is pointless to wait several minutes to try a whole EPROM only to find something is wrong.

Driving Program.

The program to drive the board is best placed in an EPROM itself, at some convenient address in the system, but of course this cannot be done until you have an EPROM programmer already working.

A listing is given to permit the job to be done temporarily in BASIC. (Two listings in fact – one in NIBL BASIC for SC/MP and the other in Tiny BASIC for the Z80). They can be used as they stand but if only machine code is available the BASIC listings may help you if you want to follow the same method in writing your own routine.



THE HOLES ABOVE ARE ADDITIONAL TO THOSE ALREADY PROVIDED ON THE STANDARD 1/8" THICK INTERNATIONAL SIZE CARD FRONT.

NOTE: DIMENSIONS SHOWN ARE FOR R.S. STYLE CARD FRONTS. (OLD TYPE). NEW STYLE CARD FRONTS ARE DIFFERENT.

DRILL 4 'A' HOLES 1·6 mm dia.
1 'B' HOLE 6·4 mm dia.

Drwn	D.M.P
Date	12-11-80
Scale	1:1

Greenbank Electronics

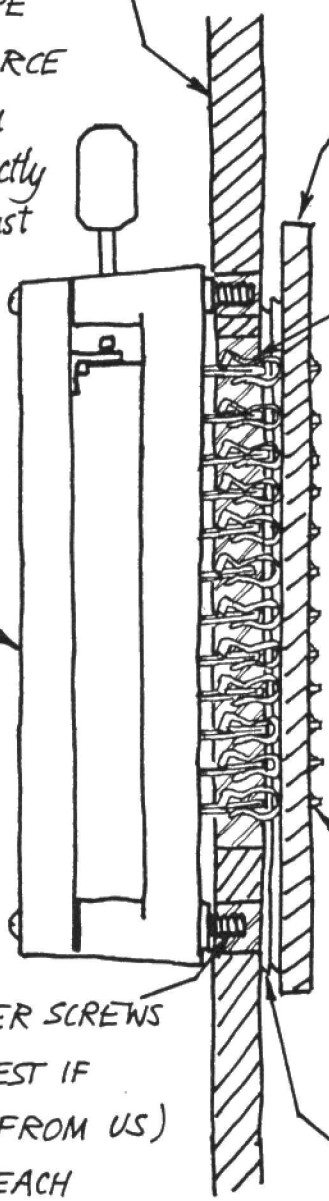
CUTTING DETAILS TO SUIT 2 'TEXTOOL' 24 PIN ZIF SOCKETS AND 10·2" LED

OUR LEAFLET AN-C24 | SHEET 12

STANDARD
2" CARD FRONT (ALUMINIUM PANEL)
DRILLED AND
CUT OUT FOR SOCKETS

TWO OFF
24-PIN
'TEXTOOL' LEVER TYPE
ZERO INSERTION FORCE
SOCKETS (Note if you are ever soldering directly to the pins the lever must be moved to the 'open' position)

0·1" PITCH 'VEROBOARD'
WITH COPPER STRIPS
(E.G. V-Q 'DIPBOARD)



24 'SOLDERCON' PINS FOR EACH SOCKET (EACH PIN IS TURNED THROUGH 90° FROM ITS NORMAL DIRECTION BEFORE SOLDERING, USE THE SOCKET ITSELF AS A JIG).

CONNECTIONS TO MAIN CIRCUIT BOARD MAY CONVENIENTLY MADE WITH 'RIBBON' CABLE

EVEN WITH THE LONGER SCREWS FITTED (FREE ON REQUEST IF SOCKET IS PURCHASED FROM US) THE SCREWS DO NOT REACH THROUGH THE PANEL - THE HOLES ARE DRILLED SIMPLY TO AID LOCATION OF SOCKET WHEN IT IS BEING MOUNTED ON THE PANEL

FIT TWO LAYERS OF DOUBLE-SIDED FOAM TAPE BETWEEN THE 'VEROBOARD' AND THE PANEL TO ACHIEVE CORRECT SPACING

(An alternative method of mounting is to remove the two screws holding the socket together, carefully disassemble it, and drill and fix the base to the panel; then replace the top and top screws.)

Drwn	D.M.P.
Date	12-11-80
Scale	2:1

Greenbank Electronics

SUGGESTED MOUNTING METHOD FOR 'TEXTOOL' ZIF SOCKETS

OUR LEAFLET AN-C24 | SHEET 13

Note that the programming time will be much longer when BASIC is used, than would be the case if an efficient machine code program had been used instead.

Also the programs show how the whole EPROM would be programmed, but it is easy to change so that even one byte at a time can be programmed if you wish.

It would be advisable to begin the program with a test that the EPROM is clear (ie. contains #FF, or decimal 255, in all locations), and to finish by reading the EPROM to ensure it has the correct data in each location, but these features are perhaps 'luxuries' at the early, test, stages of checking out the wiring.

It is worth bearing in mind that when reading the EPROM it should be read twice and the first data discarded. The reason for this is that, due to the method of latching the address at the beginning of the read or write strobe, there is not always enough time to meet the access time requirements of the EPROM before the end of the strobe, when valid data is required. Subsequent accesses at the same address are bound to return valid data, since the latched addresses will have been valid for a relatively long time.

It is only at line 100, the end of the program that the data at the control byte address `C' is returned to '0', the safe condition, so the program should not be interrupted while it is running - it might stop with the +25V connected and the 50 ms pulse stuck 'on'. (A machine code program would disable interrupts, or return the control byte to '0' before servicing them).

Another area where the use of BASIC is not ideal is in the generation of the 50 ms programming pulse.

Line 70 in the NIBL program, with its strange instruction, defines the 50 ms pulse, and lines 70, 71, 72, 75 perform the same purpose in the Z80 Tiny BASIC program. These are very system-dependent: the NIBL program needs a SC/MP II running at 4·0 MHz with no wait states, and the Tiny BASIC program needs a 2·0 MHz Z80, again with no wait states.

Systems different to those described above will need 'fine-tuning' by varying line 70 (and any others less than 80 which follow) to find a suitable instruction or set of instructions which cause a 50 ms programming pulse in your particular system. Observe the pulse with an oscilloscope when 'programming' an empty socket.

The delay can be built into the program when it is translated into machine code. The SC/MP 'delay' instruction is useful here, but some suitable sequence of instructions, or the use of a Z80-CTC chip will have to be used for the Z80 and other processors.

A slightly better solution would be to include a 50ms monostable on the PP-2 board, and either use this to activate the ISBUS 'wait' line or issue an interrupt, or make the program read the state of the monostable and loop until it 'times out'. This solution has not been described here in greater detail as it was felt that the minimum number of hardware modifications would be most helpful to most users — they can always refine the circuit once it is working.

D.M.P 14-11-80

Demonstration EPROM-Programming Program in NIBL BASIC for SC/MP.

```
 LIST
10 PR"INPUT YOUR SOURCE"
20 PR"USE '#' FOR HEX INPUT"
30 INPUT S
40 C=#D0F0 :@C=3
45 FOR J=0 TO #7FF
50 @(#A000+J)=@(S+J)
60 @C=7
70 Y=Y*#1 :REM THIS IS A 50 MS DELAY
80 @C=3
85 PR J;
90 NEXT J
100 @C=0:PR"DONE"

>
```

Demonstration EPROM-Programming Program in Z80 Tiny BASIC Language.

```
LIST
  10 INPUT'INPUT SOURCE (DECIMAL)',S
  40 C=240;OUT 3,C
  45 J=0
  50 POKE(PEEK(S+J)),(-24576+J)
  60 OUT  7,C
  70 A=0
  71 A=A+1;A=A+0
  72 REM THIS IS A DELAY OF ABOUT 50 MS (2 MHZ, NO WAIT STATES)
  75 IF A<2 GOTO 71
  80 OUT  3,C
  85 PRINT J,
  90 J=J+1; IF J<2048 GOTO 50
 100 OUT  0,C; PRINT 'DONE'
READY
>
```

Of course nowadays you'd use ZYMON's "BURN" command!